Math 3134 Final Exam Solutions

May 9, 2005

1. (15 pts)

   (a) Let S be a set. Define what an equivalence relation on S is. Make sure to explain what each of the required properties mean.

   An equivalence relation $\sim$ on S is a relation which is
   Reflexive: $x \sim x$ for all $x \in S$,
   Symmetric: if $x, y \in S$ and $x \sim y$ then $y \sim x$,
   Transitive: if $x, y, z \in S$ and $x \sim y$ and $y \sim z$ then $x \sim z$.

   (b) Define what a directed simple graph is.

   A directed simple graph is a pair $G = (V, E)$ where V is a nonempty set called the set of vertices and $E \subseteq V \times V$ is a set of ordered pairs called edges such that $(v, v) \notin E$ for any $v \in V$.

   (c) Let G be a directed (multi)graph. For vertices $U, V \in V_G$, let $U \sim V$ whenever there is a path from U to V. Is $\sim$ an equivalence relation? If so, prove it, if not, find a counterexample.

   This is not an equivalence relation because it is in general not symmetric. For example in the graph below $U \sim V$, but $V \not\sim U$.

2. (15 pts)

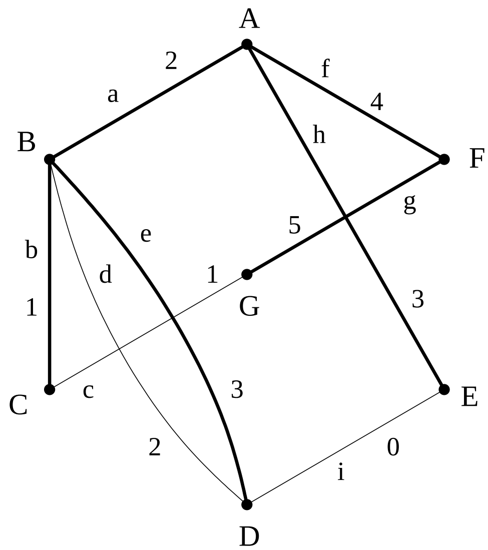
   (a) Let G be a finite, undirected graph. Define what a spanning tree of G is.

   A spanning tree T is a subgraph of G which contains all vertices of G and is a tree, that is it is connected and contains no nontrivial cycles.

   (b) Let G be a finite, undirected, weighted graph with nonnegative weights. Write down a version of Prim's algorithm which finds a <u>maximal</u> spanning tree of G if there is one, and says so if there is not. Make sure your loop terminates.

   1. Let V be a random vertex in G, and set $P = \{V\}$ and $E = \emptyset$.
   2. While there is an edge from a vertex in P to a vertex not in P
      2.1. Among all edges with one end in P and the other outside P choose e with maximal weight
      2.2. Let V be the endpoint of e that is not in P, and $E := E \cup \{e\}$, $P := P \cup \{V\}$.
      end while
   3. If $P = V_G$ then return (P, E),
      otherwise print "There is no maximal spanning tree because G is unconnec ted."
      end if

   (c) Use your algorithm starting at vertex A to find a maximal spanning tree of the graph below. Be sure to indicate the order in which the vertices and edges of the spanning tree are selected.



The vertices and edges are selected in the order below:

A, F, G, E, B, D, C

f, g, h, a, e, b (or f, g, h, a, e, c)

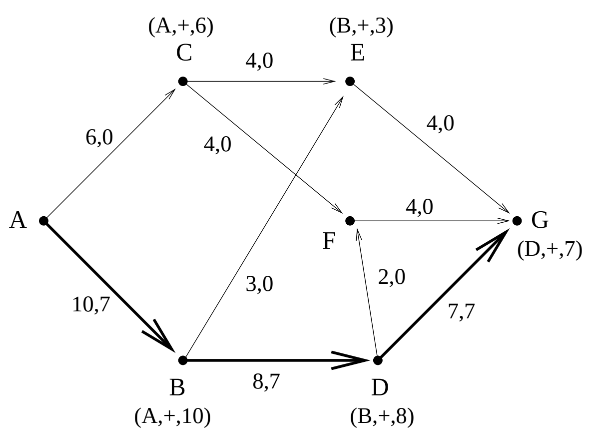
3. (10 pts)

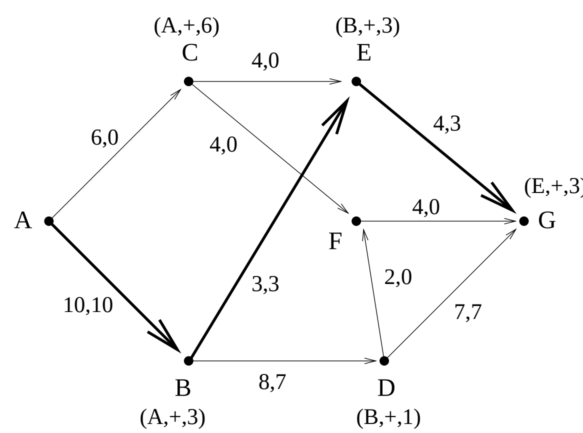
(a) Define what a transportation network means.

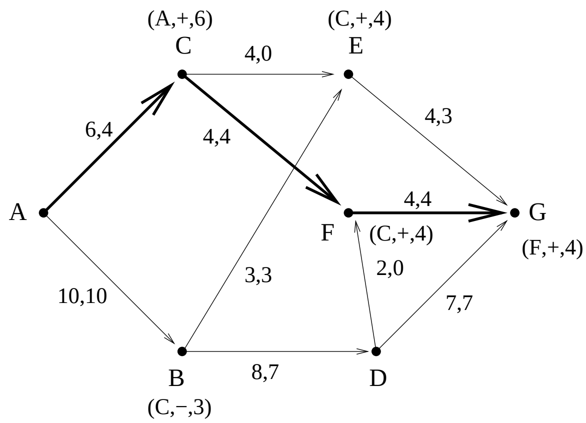
A transportation network is a finite, directed, weighted (multi)graph N with weight function $c : E_N \to \mathbb{R}$ called the capacity such that

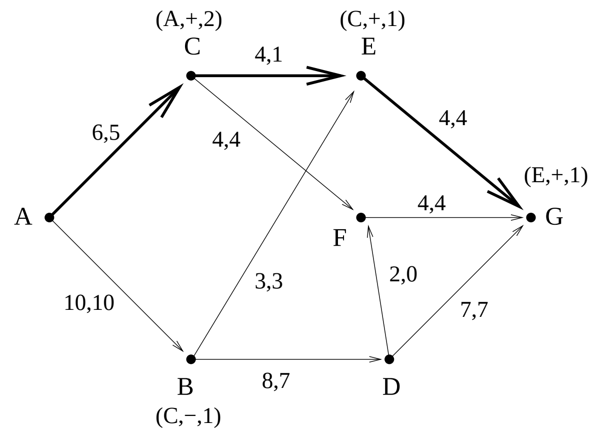
1. $c(e) \geq 0$ for all $e \in E_N$,
2. there is exactly one vertex of indegree 0 called the source,
3. there is exactly one vertex of outdegree 0 called the sink.

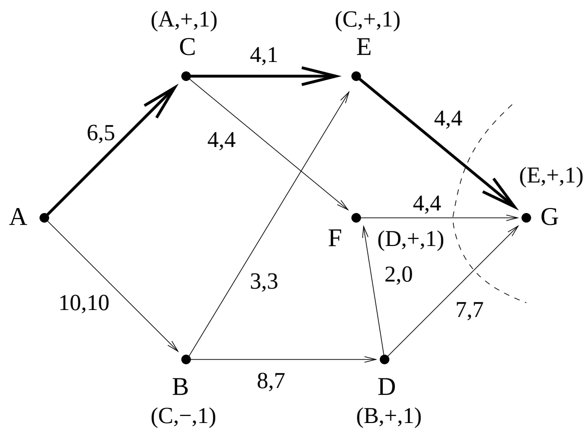
(b) Use the Augmenting Flow Algorithm to find a maximal flow and a minimal cut for the graph below. In each step, clearly label the current flow and indicate which flow augmenting path you are using. (Several copies of the graph are provided for your convenience. You may not need to use all of them.)











4. (15 pts)

(a) Let N be a network. Define what a flow on N is.

A flow f on N is a map $f : E_N \to \mathbb{R}$ such that

1. $0 \leq f(e) \leq c(e)$ for all $e \in E_N$,
2. at each vertex V, other than the source and the sink, the sum of the flows along the incoming edges is equal to the sum of the flows along the outgoing edges.

(b) Let N be a transportation network and $d > 0$. Define N' as the network with the same underlying directed graph as N, but with the capacities on all edges multiplied by d. If f is a flow on N, define $f' = df$ by $f'(e) = df(e)$ for all edges. Prove that f is a maximal flow on N if and only if f' is a maximal flow on N'. (Hint: for f' to be a maximal flow, it has to be a flow first.)

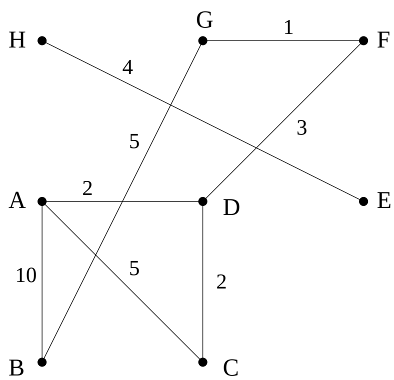
Let c' be the capacity function of N', that is $c'(e) = dc(e)$. Let f be any flow on N. Since $d > 0$,

$$0 \leq f(e) \leq c(e) \implies 0 \leq df(e) \leq dc(e) \implies 0 \leq f'(e) \leq c'(e).$$

Also, since we multiply the flow along every edge by the same scalar, the resulting f' still has the property that the sum of the incoming flows is equal to the sum of the outgoing flows at every vertex V other than the source and the sink. So f' is a flow on

N'. A symmetric argument with $1/d$ instead of d shows that if f' is a flow on N', then $f = f'/d$ is a flow on N. Therefore f is a flow on N iff $f' = df$ is a flow on N'.
Notice that since the flows out of the source are all multiplied by the same scalar d, $|f'| = d|f|$.
Now, suppose that f is a maximal flow on N. Let g' be any flow on N'. We know that $g = g'/d$ is a flow on N by the previous proof. By maximality of f, $|f| \geq |g|$. Since $d > 0$, we can multiply both sides by d to get $|f'| = d|f| \geq d|g| = |g'|$. Hence f' is a maximal flow on N'. By a symmetric argument, if f' is a maximal flow on N', f is a maximal flow on N. So f is a maximal flow on N iff f' is a maximal flow on N'.

5. (10 pts) Use Dijkstra's algorithm to find shortest distances from vertex A to every other vertex in the graph below.
   (a) Fill in the table with the labels at every intermediate step. (You may not need to use all of the rows.)



Loop	A	B	C	D	E	F	G	H
1	$0, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$
2	$0, -$	$10, A$	$5, A$	$2, A$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$
3	$0, -$	$10, A$	$4, D$	$2, A$	$\infty, -$	$5, D$	$\infty, -$	$\infty, -$
4	$0, -$	$10, A$	$4, D$	$2, A$	$\infty, -$	$5, D$	$\infty, -$	$\infty, -$
5	$0, -$	$10, A$	$4, D$	$2, A$	$\infty, -$	$5, D$	$6, F$	$\infty, -$
6	$0, -$	$10, A$	$4, D$	$2, A$	$\infty, -$	$5, D$	$6, F$	$\infty, -$
7	$0, -$	$10, A$	$4, D$	$2, A$	$\infty, -$	$5, D$	$6, F$	$\infty, -$
8								
9								

   (b) What shortest path from A to G did the algorithm find?

   A, D, F, G

   (c) What shortest path from A to H did the algorithm find?

   None. There exists no path from A to H.

6. (5 pts) Prove that

$$\sum_{k=0}^{n} (k+1) \binom{n}{k} = 2^n + n2^{n-1}.$$

(Hint: Differentiate $x(x+1)^n$ once expanding it first using the Binomial Theorem and once without expanding.)

$$\begin{aligned}
x(x+1)^n &= x\sum_{k=0}^{n}\binom{n}{k}x^k \\
x(x+1)^n &= \sum_{k=0}^{n}\binom{n}{k}x^{k+1} \\
\frac{d}{dx}x(x+1)^n &= \frac{d}{dx}\sum_{k=0}^{n}\binom{n}{k}x^{k+1} \\
(x+1)^n + nx(x+1)^{n-1} &= \sum_{k=0}^{n}(k+1)\binom{n}{k}x^k \\
\left[(x+1)^n + nx(x+1)^{n-1}\right]_{x=1} &= \left[\sum_{k=0}^{n}(k+1)\binom{n}{k}x^k\right]_{x=1} \\
2^n + n2^{n-1} &= \sum_{k=0}^{n}(k+1)\binom{n}{k}
\end{aligned}$$

7. (10 pts) **Extra credit problem.** Let n and k be positive integers. A partition of n into k terms is a set of k positive integers $\{n_1, n_2, \ldots, n_k\}$ such that $n_1 + n_2 + \cdots + n_k = n$. (This has nothing to do with partitions of sets.) Note that since the n_i are elements of a set, not a list, their order does not matter. For example, $1+2+3$ and $2+2+2$ are different partitions of 6, but $1+2+3$ and $3+2+1$ are the same. Find (and justify) a formula for the number of different partititions of n into k terms.

   Note: This problem is harder than I had intended. If the order of the parts mattered, then you could get the answer by similar logic to what we used to find the number of ways of picking n elements out of a set containing k elements with replacement and without regard to the order. Recall that this argument reduced to counting the number of ways of permuting n dots and $k-1$ bars. On first approach, you could represent the number n by n dots and then you could insert $k-1$ bars to divide it into k parts. But the problem is that this could result in two bars being next each other giving you a part that is 0, which is not allowed. So what you want to do is permute $n-k$ dots and $k-1$ bars, then add exactly one dot to each part to make sure it is positive. As we argued in class, the number of ways of permuting $n-k$ dots and $k-1$ bars is $\frac{(n-k+k-1)!}{(n-k)!(k-1)!} = \binom{n-1}{k-1}$.

   Unfortunately, the way the problem is stated (due to temporary confusion on my part while writing up the exam), the order of the parts does not matter, so the above process will give different arrangements of dots and bars that correspond to the same partition. For example, $\cdots|\cdot\cdot|\cdot$ and $\cdot\cdot|\cdots|\cdot$ would both correspond to $3+2+1$. On first thought, you may be tempted to say that each partition corresponds to $k!$ different arrangements of the dots and bars because we can permute its k parts in $k!$ ways. While this logic works fine for $1+2+3$, in which you could list the three different terms in $3! = 6$ ways, it fails for $2+2+2$ because no matter how you permute the three $2s$, you always get the same thing.

   In fact, I don't know of a closed formula for the solution. The best we can do is give a recursive formula, similar to Pascal's Triangle. Let $f(n,k)$ be the number of different partitions of n into k *or fewer* parts. Then the number of ways to partition n into exactly k parts is $f(n,k) - f(n,k-1)$.

Now, take a partition of n into k parts $n_1+n_2+\cdots+n_k$. Subtract 1 from each part to get $(n_1-1)+(n_2-1)+\cdots+(n_k-1)$. Some of these terms could be 0. Throw these away, and you get a partition of $n-k$ into k *or fewer* terms. (If all of the terms are 0, then you just get $0=0$, which we will call the only partition of 0 so that this logic does not fail.) In fact, if I handed you a partition of $n-k$ into k or fewer terms, you could recover which partition of n into k terms it came from by adding enough 0s to it to have k terms, and then adding 1 to each term. So we have a one-to-one correspondence between partitions of n into k terms and partitions of $n-k$ into $\le k$ terms. We just proved

$$f(n,k)-f(n,k-1)=f(n-k,k)$$
$$f(n,k)=f(n,k-1)+f(n-k,k)$$

Obviously, $f(n,1)=1$, $f(0,k)=1$ if $k\ge 1$ (remember 0 has only one partition), and $f(n,k)=0$ if $n<0$. You can now use the recursive formula to compute $f(n,k)$ for any values of $n,k\in\mathbb{N}$.

For example, let's compute how many ways we have of partitioning 6 into 3 parts. This is $f(6,3)-f(6,2)$. By the above

$$f(6,3)-f(6,2)=f(3,3)=f(3,2)+f(0,3)=f(3,2)+1$$
$$f(3,2)=f(3,1)+f(1,2)=1+f(1,1)+f(-1,2)=1+1=2$$
$$f(6,3)-f(6,2)=3$$

The three partitions are $4+1+1=3+2+1=2+2+2$.

I apologize that this turned out to be much more complicated than I had intended. It was encouraging to see that some of you still made progress on this problem.

Looking on the bright side, if you read and understood the solution here, you have certainly learned something new and interesting. In fact, if we had had time to discuss generating functions, we could have solved this problem that way.

8. (10 pts) **Hard extra credit problem.** Let G be a finite, undirected, simple graph, and let $\chi(G)$ be the chromatic number of G. Prove that $\chi(G)\le k$ if and only if the edges of G can be directed so that a longest path has length at most $k-1$.

First suppose that $\chi(G)\le k$. This means you can color the vertices of G with no more than k colors so that no adjacent ones have the same color. Call these colors $1,2,\ldots,m$ where $m\le k$. Now direct the edges so that each edge points from smaller number to larger number. Since no edge connects two identical numbers, this is possible to do. So the numbers on the vertices along any path are increasing. The largest number is $m\le k$, therefore no path can be longer than k vertices, or $k-1$ edges. So a longest path has length at most $k-1$.

Now assume that the edges can be directed so that a longest path has length at most $k-1$. For each vertex V, let $f(V)$ be the length of a longest path starting at that vertex. Notice $0\le f(V)\le k-1$. Let U and V be adjacent vertices in G. Suppose, without loss of generality, that the edge between U and V is directed so that it points from U to V. Then $f(U)\ge f(V)+1$ because you could always take a path from U to V and then continue along a longest path starting at V. In particular, $f(U)\ne f(V)$ for adjacent vertices U and V. Now assign $f(V)$ to V as its color, and you have a coloring of G with at most k colors.